Package 'OvRSeq'

November 3, 2023

Title Ovarian Cancer RNA-Seq Analysis Package

Version 0.1.1

Description
OvRSeq is an R package for analyzing RNA sequencing data from ovarian cancer patients. The package includes functions for quality control, normalization, differential gene expression analysis,
and pathway analysis. OvaRSeq also provides options for visualization of results, such as heatmaps and volcano plots.
The package is designed to be user-friendly and applicable to various types of RNA sequencing data.

License MIT

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0),
biomaRt,
SummarizedExperiment,
EnsDb.Hsapiens.v86

Imports caret,
biomaRt,
SummarizedExperiment,
EnsDb.Hsapiens.v86

RoxygenNote 7.2.3

R topics documented:

avg_expression_for_signature_se 2
brcaness_classifier 3
brcaness_signature 4
calculateIPS 5
classify_brcaness 5
deconvolute_immune 6
getEnsemblIds 7
getGeneLength 8

get_consensus_ov_subtypes 8
hello 9
immune_signatures 10
immune_signature_score 11
IPS_genes 11
load_brcaness_classifier 12
load_brcaness_signature 12
load_immune_signatures 13
load_TCGA_OV 14
log2Norm 14
OvaRSeqDataSet 15
OvaRSeqDataSet-class 15
RPKMNorm 16
ssGSEA_OV_custom 16
TCGA_OV 17
TPMNorm 18
train_rf 19
tumor_immune_phenotype_signature 20
update_se_with_entrez_ids 21

Index **22**

`avg_expression_for_signature_se` *Compute average expression values for gene signatures and enrich colData of a SummarizedExperiment object*

Description

Given a 'SummarizedExperiment' object and a matrix or data frame of gene signatures, this function computes the average expression values for each gene signature and enriches the 'colData' of the 'SummarizedExperiment' object with the computed values.

Usage

```
avg_expression_for_signature_se(se, gene_sig, genesetname)
```

Arguments

`se`	A 'SummarizedExperiment' object with gene expression data.
`gene_sig`	A matrix or data frame where each column represents a gene signature and each cell contains a gene symbol. Empty cells should be represented as empty strings.
`genesetname`	A character string that serves as a prefix for the new columns added to the 'colData' of the 'SummarizedExperiment' object.

Value

A 'SummarizedExperiment' object enriched with new columns in 'colData', each representing the average expression value of a given gene signature for each sample.

Examples

```
# Assuming 'se' is a SummarizedExperiment object and 'gene_sig' is your gene signatures matrix
enriched_se <- avg_expression_for_signature_se(se, gene_sig, "MyGeneSet")
```

---

brcaness_classifier *Random Forest Classifier for BRCAness*

Description

This data object contains a trained random forest classifier that utilizes a 24-gene expression signature to classify patients into two categories: those with BRCAness and those with BRCA status. The classifier was trained using multiomics data, including RNA-Seq data, from the TCGA-OV dataset.

Usage

```
data(brcaness_classifier)
```

Format

A list containing a trained random forest classifier object.

Details

A trained random forest classifier for predicting BRCAness status using a 24-gene expression signature. This classifier was trained with multiomics data from the TCGA-OV dataset and utilizes RNA sequencing (RNA-Seq) data to classify patients as either having BRCAness or BRCA status.

The random forest classifier in this data object was trained with feature selection to optimize its ability to predict BRCAness status based on RNA-Seq data. It is a result of a multiomics approach aimed at accurately classifying patients.

Examples

```
# Load the trained BRCAness classifier
data(brcaness_classifier)

# Predict BRCAness status for a new patient using the classifier
new_patient_data <- ...  # Replace with new patient's RNA-Seq data
prediction <- predict(brcaness_classifier, new_patient_data)
```

brcaness_signature *BRCAness Gene Signature*

Description

The BRCAness gene signature is a result of an extensive feature selection process using machine learning models. It represents a subset of genes that are critical for discriminating between BRCAness and non-BRCAness samples based on gene expression data from the TCGA-OV dataset.

Usage

```
data(brcaness_signature)
```

Format

A vector containing the gene symbol for 24 genes.

Details

A gene signature developed through feature selection to optimally classify TCGA-OV samples into BRCAness and non-BRCAness categories. The signature consists of gene expression values (2TPM+1 normalized) for 24 genes. These genes were selected based on their importance in three different machine learning models (Random Forest, Ada Boost, and Gradient Boosting) after recursive feature elimination. Only genes that were among the top 50 most important features in at least two of the three models were included in this signature.

See Also

TCGA_OV for the dataset used for feature selection and training.

Examples

```
# Load the BRCAness gene signature
data(brcaness_signature)

# Access gene expression values for the first gene
brcaness_signature
```

`calculateIPS` *Calculate Immunophenoscore (IPS) from a SummarizedExperiment*

Description

This function calculates the Immunophenoscore (IPS) and its components scores from gene expression data encapsulated in a 'SummarizedExperiment' object. It requires a specific set of genes and corresponding weights provided in a separate file.

Usage

```
calculateIPS(se)
```

Arguments

`se`	A 'SummarizedExperiment' object containing normalized gene expression data.

Value

A data frame with samples as rows and calculated scores (WG, MHC, CP, EC, SC, MDSC, TREG, AZ, IPS) as columns.

References

Immunophenogram: This R-script can be used to calculate Immunophenoscore (IPS) and generate Immunophenogram from "EXPR.txt" and "IPS_genes.txt". The script and associated documentation can be found at the following URL: https://github.com/icbi-lab/Immunophenogram (C) ICBI, Medical University of Innsbruck, Biocenter, Division of Bioinformatics Version 1.0 08.07.2016

`classify_brcaness` *Classify Samples Using BRCAness Classifier*

Description

This function uses the BRCAness classifier to predict the BRCAness status of the samples in a SummarizedExperiment object. It first checks that all the genes in the BRCAness signature are present in the count data of the SummarizedExperiment object before making predictions.

Usage

```
classify_brcaness(se, brcaness_classifier, brcaness_signature)
```

Arguments

`se`	A SummarizedExperiment object with count data.
`brcaness_classifier`	A random forest classifier object trained on BRCAness samples.
`brcaness_signature`	A vector of gene symbols representing the BRCAness gene signature.

Value

A SummarizedExperiment with BRCAness predictions, in ColData.

`deconvolute_immune` *Deconvolute Immune Cell Fractions from Gene Expression Data*

Description

This function deconvolutes immune cell fractions from gene expression data using methods from the ‘immunedeconv‘ package.

Usage

```
deconvolute_immune(se, method)
```

Arguments

`se`	A Summarized Experiment of gene expression values (logTPM+1), where rows are genes and columns are samples.
`method`	A character string indicating the deconvolution method to be used. See details for available methods.

Details

Available methods and their respective licensing and citations are as follows:

method	license	citation
quanTIseq	free (BSD)	Finotello et al. (2019) Genome Medicine
TIMER	free (GPL 2.0)	Li et al. (2016) Genome Biology
CIBERSORT	free for non-commerical use only	Newman et al. (2015) Nature Methods
MCPCounter	free (GPL 3.0)	Becht et al. (2016) Genome Biology
xCell	free (GPL 3.0)	Aran et al. (2017) Genome Biology
EPIC	free for non-commercial use only	Racle et al. (2017) ELife

Note: While ‘immunedeconv‘ itself is free (BSD), you may need to obtain a license to use individual methods. Please ensure you cite both this package and the method(s) you use in your work.

Value

A list with deconvolution results.

References

Sturm, G., et al. (2019) Bioinformatics. https://doi.org/10.1093/bioinformatics/btz363

Examples

```
# data <- ... # Your gene expression data (logTPM+1)
# res <- deconvolute_immune(data, method="timer")
```

`getEnsemblIds` *Get Ensembl IDs from Gene Symbols*

Description

Takes a vector of gene symbols and returns a data frame of Ensembl gene IDs and gene symbols using the biomaRt package.

Usage

```
getEnsemblIds(gene_symbols, org = "hsapiens_gene_ensembl")
```

Arguments

`gene_symbols`	A character vector of gene symbols to look up
`org`	A string specifying the organism (default: "hsapiens_gene_ensembl")

Value

A data frame with two columns: "ensembl_gene_id" and "external_gene_name"

Examples

```
getEnsemblIds(c("BRCA1", "TP53"))
```

getGeneLength *Retrieve gene length from Ensembl IDs*

Description

This function retrieves the length of genes based on their Ensembl IDs. The gene length is defined as the sum of the lengths of all exons of the gene. The function uses the `exonsBy` function from the **ensembldb** package to obtain exon information, and the **EnsDb.Hsapiens.v86** package for the human reference database.

Usage

```
getGeneLength(ensembl_ids, org = "hsa")
```

Arguments

`ensembl_ids`	A data frame containing two columns: `ensembl_gene_id` and `external_gene_name`. The former contains the Ensembl gene IDs of the genes for which the gene length should be retrieved, while the latter contains their corresponding external gene symbols.
`org`	A character string indicating the organism for which gene length should be retrieved. The default value is "hsa", corresponding to the human genome.

Value

A data frame containing three columns: `ensembl_gene_id`, `external_gene_name`, and `length`. The length column contains the length of each gene in base pairs (bp).

---

get_consensus_ov_subtypes

Obtain Consensus Subtypes of High-Grade Serous Ovarian Cancer

Description

This function gets the consensus molecular subtypes of high-grade serous (HGS) ovarian cancer using the ‘consensusOV‘ package. The function implements a consensus classifier which consolidates and improves on the robustness of proposed subtype classifiers.

Usage

```
get_consensus_ov_subtypes(
  se,
  ids_type,
  concordant.tumors.only = TRUE,
  remove.using.cutoff = FALSE,
  percentage.dataset.removed = 0.75
)
```

Arguments

se	A Summarized Experiments of gene expression values, with gene symbol or entrez id as identifiers.
ids_type	A character string indicating the type of IDs used in the 'data' row names. Can be either "symbol" or "entrez".
concordant.tumors.only	Logical. If TRUE, only tumors that are concordantly classified across all datasets are included in the final classification. Defaults to TRUE.
remove.using.cutoff	Logical. If TRUE, tumors with poor classification confidence are removed using the optimal cutoff. Defaults to FALSE.
percentage.dataset.removed	Numeric value indicating the percentage of the dataset to be removed based on classification confidence. Defaults to 0.75.

Value

Summarized Experiments with Tumor_Molecular_Subtypes of consensus molecular subtypes for the samples, in colData.

References

Chen G, Kannan L, Geistlinger L, Kofia V, Safikhani Z, Gendoo D, Parmigiani G, Birrer M, Haibe-Kains B, Waldron L (2018). "Consensus on molecular subtypes of high-grade serous ovarian carcinoma." Clinical Cancer Research, 24, 4990. doi:10.1158/1078-0432.CCR-18-0784.

Examples

```
load_TCGA_OV
TCGA_OV <- get_consensus_ov_subtypes(TCGA_OV, ids_type = "symbol")
```

---

hello *Hello, World!*

Description

Prints 'Hello, world!'.

Usage

```
hello()
```

Examples

```
hello()
```

`immune_signatures` *Immune Signatures*

Description

The 'immune_signatures' object contains well-defined immune-related gene signatures in GMT format. These signatures are valuable for characterizing immune processes and activities in biological data, including gene expression profiles.

Usage

```
data(immune_signatures)
```

Format

A list of lists, where each sublist contains: - 'Name': The name of the immune signature. - 'Genes': A character vector of gene symbols representing the signature genes.

Details

A GMT (Gene Matrix Transposed) file containing well-defined immune-related gene signatures. Each signature consists of a list with the name of the signature and a vector of gene symbols representing the genes that constitute the signature. These signatures are used to characterize immune-related processes, including T cell inflammation, IFN gamma signature, cytolytic activity, and cytotoxic T lymphocyte function.

See Also

For more information on GMT files and gene set enrichment analysis (GSEA), refer to the relevant literature and software documentation.

Examples

```
# Load the immune signatures
data(immune_signatures)

# Access the genes in the "T cell inflammation" signature
t_cell_inflammation_genes <- immune_signatures$T_cell_inflammation$Genes
```

immune_signature_score *Calculate enrichment scores for immune signatures*

Description

This function calculates enrichment scores for a list of immune signatures using the GSVA method. The user can provide their own gene list or use one of the pre-defined ones. The function returns an updated SummarizedExperiment object with the enrichment scores added to the colData.

Usage

```
immune_signature_score(se, method = "ssgsea", genelist)
```

Arguments

se	A SummarizedExperiment object containing RNA sequencing data
method	A character string indicating the method to use for calculating enrichment scores. Default is "ssgsea".
genelist	A character vector with the gene list for the immune signature.

Value

A SummarizedExperiment object with the enrichment scores added to the colData.

IPS_genes *Immunophenoscore (IPS) Genes and Weights*

Description

A dataset containing genes and corresponding weights used to calculate the Immunophenoscore (IPS), which is a measure of the immune landscape of a tumor.

Usage

```
data(IPS_genes)
```

Format

A data frame with the following columns:

GENE (character) Official gene symbol.

WEIGHT (numeric) Weight of the gene in the IPS calculation.

CATEGORY (character) The immunological category to which the gene belongs (e.g., MHC, CP, EC, SC, MDSC, TREG, AZ).

Source

Immunophenogram project: https://github.com/icbi-lab/Immunophenogram

Examples

```
# Load the IPS genes and weights
data(IPS_genes)

# Access the gene symbols and weights
ips_genes <- IPS_genes$GENE
ips_weights <- IPS_genes$WEIGHT
```

`load_brcaness_classifier`

Load BRCAness Classifier

Description

Loads the pre-trained BRCAness classifier from the "brcaness_classifier.rda" data file and returns it as the ‘brcaness_classifier‘ object.

Usage

```
load_brcaness_classifier()
```

Value

The pre-trained BRCAness classifier as a list.

Examples

```
# Load the BRCAness classifier
brcaness_classifier <- load_brcaness_classifier()
```

`load_brcaness_signature`

Load BRCAness Gene Signature

Description

Loads the BRCAness gene signature from the "brcaness_signature.rda" data file and returns it as a vector with gene symbol for the for 24 genes.

Usage

```
load_brcaness_signature()
```

Value

A vector with gene symbol for the BRCAness gene signature.

Examples

```
# Load the BRCAness gene signature
brcaness_signature <- load_brcaness_signature()

# Access gene expression values for the first gene
brcaness_signature$Gene1
```

---

load_immune_signatures

Load Immune Signatures

Description

Loads the immune signatures from the "immune_signatures.rda" data file and returns them as a list of lists, where each sublist contains the name of an immune signature and a vector of gene symbols representing the genes that constitute the signature.

Usage

```
load_immune_signatures()
```

Value

A list of lists, where each sublist contains: - ‘Name‘: The name of the immune signature. - ‘Genes‘: A character vector of gene symbols representing the signature genes.

Examples

```
# Load the immune signatures
immune_signatures <- load_immune_signatures()

# Access the genes in the "T cell inflammation" signature
t_cell_inflammation_genes <- immune_signatures$T_cell_inflammation$Genes
```

load_TCGA_OV *Load TCGA-OV Dataset*

Description

Loads the TCGA-OV dataset from the "TCGA_OV.rda" file and returns it as a SummarizedExperiment object.

Usage

```
load_TCGA_OV()
```

Value

A SummarizedExperiment object containing raw RNA-Seq counts from the TCGA-OV dataset with associated metadata columns.

Examples

```
# Load the TCGA-OV dataset
my_dataset <- load_TCGA_OV()

# Access metadata columns
metadata(my_dataset)$AGE
metadata(my_dataset)$TUMOR_GRADE
```

log2Norm *Log2 normalization of count data*

Description

This function performs log2 normalization of count data. It adds 1 to each count value and takes the logarithm base 2 of the resulting value.

Usage

```
log2Norm(count_data)
```

Arguments

count_data	A matrix of count data with genes in rows and samples in columns..

Value

A matrix of count data with genes in rows and samples in columns, containing the log2 normalized count data.

Examples

```
data("example_counts")
data("example_gene_length")
se <- SummarizedExperiment::SummarizedExperiment(assays = list(counts = example_counts))
log2_norm_data <- log2Norm(se)
```

`OvaRSeqDataSet` *OvaRSeq constructor*

Description

OvaRSeq constructor

Usage

```
OvaRSeqDataSet(countData, colData)
```

Arguments

`countData`	a matrix or data frame contains gene count
`colData`	a `DataFrame` or `data.frame`
`...`	optional arguments passed to `SummarizedExperiment`

Value

a OvaRSeq object

`OvaRSeqDataSet-class` *OvaRSeqDataSet class*

Description

`OvaRSeqDataSet` is a class inherited from `SummarizedExperiment`. It is used to store the count matrix, colData, and design formula in differential expression analysis.

References

Martin Morgan, Valerie Obenchain, Jim Hester and Hervé Pagès (2018). SummarizedExperiment: SummarizedExperiment container. R package version 1.12.0.

RPKMNorm *RPKM normalization of count data*

Description

Takes a count data matrix, and applies RPKM normalization.

Usage

```
RPKMNorm(count_data)
```

Arguments

count_data	A matrix of count data with genes in rows and samples in columns.

Value

A RPKM-normalized matrix with the same dimensions as count_data.

Examples

```
# Load example data from SummarizedExperiment package
data("example_se")
# Extract count data and gene length data
counts <- assay(example_se)
# Apply RPKM normalization
normalized_counts <- RPKMNorm(counts)
```

ssGSEA_OV_custom *Perform Single-Sample Gene Set Enrichment Analysis (ssGSEA) on a SummarizedExperiment*

Description

This function performs ssGSEA on a given 'SummarizedExperiment' object using custom gene sets provided in GMT format. It leverages the 'GSVA' package to compute enrichment scores for each sample in the experiment. GSVA is a non-parametric, unsupervised method for estimating variation of gene set enrichment through the samples of an expression dataset.

Usage

```
ssGSEA_OV_custom(se, gmt_list)
```

Arguments

se	A 'SummarizedExperiment' object containing the expression data.
gmt_list	A list where each element is a character vector of gene symbols representing a gene set. The names of the list elements are used as gene set names.

Value

A matrix of enrichment scores with gene sets as rows and samples as columns.

References

Hänzelmann S, Castelo R, Guinney J (2013). "GSVA: gene set variation analysis for microarray and RNA-Seq data." BMC Bioinformatics, 14, 7. <doi:10.1186/1471-2105-14-7> For a full list of citations, use 'citation("GSVA")' in R.

Examples

```
# Assuming 'se' is a SummarizedExperiment object and 'gmt_list' is your list of gene sets
enrichment_scores <- ssGSEA_OV_custom(se, gmt_list)
```

TCGA_OV *TCGA-OV RNA-Seq Dataset*

Description

This dataset contains raw RNA-Seq counts from the TCGA-OV dataset. It includes metadata columns that provide additional information about the samples, including patient age, tumor grade, clinical stage, tumor residual disease status, and BRCAness classification.

Usage

```
data(TCGA_OV)
```

Format

A SummarizedExperiment object.

Details

A SummarizedExperiment object containing raw RNA-Seq counts from the TCGA-OV dataset, along with metadata columns including AGE, TUMOR_GRADE, CLINICAL_STAGE, TUMOR_RESIDUAL_DISEASE, and BRCAness.

The TCGA-OV dataset is stored as a SummarizedExperiment object, which provides an organized and efficient structure for working with high-throughput genomics data. The metadata columns allow for the exploration of clinical and molecular attributes associated with the samples.

See Also

SummarizedExperiment for more information on SummarizedExperiment objects.

Examples

```
# Load the TCGA-OV dataset
data(TCGA_OV)

# Access metadata columns
metadata(TCGA_OV)$AGE
metadata(TCGA_OV)$TUMOR_GRADE
```

TPMNorm *TPM normalization of count data*

Description

Takes a count data matrix and a gene length data frame, and applies TPM normalization.

Usage

```
TPMNorm(count_data, gene_length)
```

Arguments

count_data	A matrix of count data with genes in rows and samples in columns.
gene_length	A vector with gene IDs as names and transcript lengths in kilobases (KB).

Value

A TPM-normalized matrix with the same dimensions as count_data.

Examples

```
# Load example data from SummarizedExperiment package
data("example_se")
# Extract count data and gene length data
counts <- assay(example_se)
gene_length <- rowData(example_se)$gene_length
# Apply TPM normalization
normalized_counts <- TPMNorm(counts, gene_length)
```

train_rf *Train a random forest model on gene expression data*

Description

This function trains a random forest model on a summarized experiment object containing gene expression data, using a specified gene signature to select a subset of genes, and a specified label in the colData to use as the target variable. The function returns the best random forest model.

Usage

```
train_rf(se, label, gene_signature)
```

Arguments

se	A SummarizedExperiment object containing gene expression data
label	A string indicating the name of the column in colData containing the label to use as the target variable for classification
gene_signature	A character vector containing the names of genes to use in the analysis

Value

A trained random forest model

Examples

```
# Load the TCGA_OV.rda and BRCAness_signature.rda files
data(TCGA_OV)
data("brcaness_signature")

# Train a random forest model using the BRCAness label
rf_model <- train_rf(se, "BRCAness", brcaness_signature)

# Print the model
print(rf_model)

# Make predictions using the model
predictions <- predict(rf_model, newdata = t(assay(TCGA_OV)[rownames(BRCAness_signature),]))

# Print the predictions
print(predictions)
```

`tumor_immune_phenotype_signature`
Tumor Immune Phenotype Signature

Description

The 'tumor_immune_phenotype_signature' object contains a gene signature that classifies ovarian cancer tumor immune phenotypes. It was developed using integrated digital pathology and transcriptome analysis, with a focus on CD8+ T cell presence and position within the tumor.

Usage

```
data(tumor_immune_phenotype_signature)
```

Format

A list containing gene symbols representing the 148 genes in the tumor immune phenotype signature.

Details

A gene signature developed based on digital pathology and transcriptome analysis to classify tumor immune phenotypes (infiltrate, excluded, desert) in ovarian cancer. This signature was derived from a classification model using 157 genes that describe the presence and position of CD8+ T cells relative to the tumor center or margin in the ICON7 cohort.

References

Desbois M, Udyavar AR, Ryner L, Kozlowski C, Guan Y, Dürrbaum M, et al. Integrated digital pathology and transcriptome analysis identifies molecular mediators of T-cell exclusion in ovarian cancer. Nat Commun. 2020;11:5583.

Examples

```
# Load the tumor immune phenotype signature
data(tumor_immune_phenotype_signature)

# Access the gene symbols in the signature
signature_genes <- tumor_immune_phenotype_signature
```

update_se_with_entrez_ids *Update Gene Symbols to Entrez IDs in a SummarizedExperiment*

Description

This function takes a SummarizedExperiment object with gene symbols and updates the row names with corresponding Entrez IDs.

Usage

```
update_se_with_entrez_ids(se)
```

Arguments

se	A SummarizedExperiment object.

Value

A SummarizedExperiment object with Entrez IDs as row names.

Examples

```
# Assuming you have a SummarizedExperiment object named 'se'
se_updated <- update_se_with_entrez_ids(se)
rownames(se_updated)
```

Index

avg_expression_for_signature_se, 2

brcaness_classifier, 3
brcaness_signature, 4

calculateIPS, 5
classify_brcaness, 5

deconvolute_immune, 6

get_consensus_ov_subtypes, 8
getEnsemblIds, 7
getGeneLength, 8

hello, 9

immune_signature_score, 11
immune_signatures, 10
IPS_genes, 11

load_brcaness_classifier, 12
load_brcaness_signature, 12
load_immune_signatures, 13
load_TCGA_OV, 14
log2Norm, 14

OvaRSeqDataSet, 15
OvaRSeqDataSet-class, 15

RPKMNorm, 16

ssGSEA_OV_custom, 16
SummarizedExperiment, *15*

TCGA_OV, 17
TPMNorm, 18
train_rf, 19
tumor_immune_phenotype_signature, 20

update_se_with_entrez_ids, 21